Overview and explanation of all processing that was done on the dataset.

1. Dropping Columns

A new DataFrame, ‘df1’, is created by dropping the ‘reservation_status_date’ column from the original DataFrame ‘df’.

2. Identifying Categorical Variables

Identifying the names of all categorical columns in ‘df1’ whose data type is the object. Identifying categorical variables is crucial because they might need different preprocessing steps, like encoding, compared to numerical variables.

3. Concatenating Lists

Adding columns with numerical representations of categorical variables to the list ‘cate’. Although some variables may be represented numerically like ‘IDs’, they could inherently be categorical in nature. Hence, they are added to the categorical list for consistent preprocessing.

4. Fill NA Values in ‘agent’ and ‘company’

Filling missing values in the ‘agent’ and ‘company’ columns with 0s. Missing values can impede many machine learning algorithms. Filling them ensures that algorithms function correctly.

5. Creating New Binary Features

Creating new binary features ‘in_company’ and ‘in_agent’ to classify rows based on whether the ‘company’ and ‘agent’ columns are 0 or not. Simplifying the dataset and creating binary features can help in understanding patterns more easily. Here, knowing if there was a company or agent involved could be directly linked to the cancellation outcome.

6. Creating the ‘same_assignment’ Feature

Creating a new binary feature, ‘same_assignment’, that is 'Yes' if the reserved room type is the same as the assigned room type, and 'No' otherwise. Feature engineering can often unveil hidden patterns in data. By comparing reserved and assigned room types, this step could help identify if discrepancies lead to cancellations.

7. Further Dropping of Columns

Dropping the columns ‘reserved_room_type’, ‘assigned_room_type’, ‘agent’, and ‘company’ from ‘df1’. After deriving new features, the original ones (from which these features were derived) might become redundant and are hence removed to reduce dimensionality.

8. Re-encoding ‘is_repeated_guest’

Re-encoding the ‘is_repeated_guest’ column with 'YES' for 1 and 'NO' for 0. Making data more human-readable or compatible with certain algorithms. By changing numbers to 'YES' and 'NO', it becomes easier to understand the nature of the data briefly.

9. Filling Missing Values in ‘country’

Filling the missing values in the ‘country’ column with the mode of that column. Similar to Step 4, handling missing values ensures smoother downstream processing. The mode (most frequent value) is a common method for imputing categorical missing values.

10. Encoding Categorical Features

Encoding the categorical features using an Ordinal Encoder. Many machine learning models require numerical input, so categorical features are encoded to numbers for compatibility.

11. Imputing Missing Values in ‘children’

Filling missing values in the ‘children’ column with its mode. Missing values in the dataset can lead to biases or errors in models. By filling in the mode, the most common trend in the data is maintained.

12. Standardizing Continuous Features

Scaling the continuous variables using Standard Scaler. Algorithms like SVM, KNN, or Logistic Regression can be sensitive to feature scales. Standardizing ensures all features have the same scale, leading to better performance and faster convergence.

13. Calculating Correlations

Calculating the correlation of all numerical columns with the ‘is_canceled’ column. Identifying how different numerical features correlate with the target variable ‘is_canceled’ can help in feature selection and understanding the most influential factors in the dataset.

14. Visualization of Correlations

Creating a horizontal bar plot to visualize the absolute correlation values between the ‘is_canceled’ column and other numerical columns. Visual representation aids in better understanding and interpreting the correlations. It helps in identifying key features that might require more attention or might be candidates for removal if they have minimal correlation.

15. Dropping Additional Column

Creating a new DataFrame, ‘df2’, by dropping the ‘reservation_status’ column from ‘df1’. The ‘reservation_status’ column might be closely related to the ‘is_canceled’ column, leading to data leakage. Removing it ensures the model doesn't get unfair information.

16.Dropping columns that are not useful

These are likely personal identifiers. Including such personal data poses privacy risks. For modeling purposes, these fields usually don't provide any predictive power, and retaining them might lead to overfitting on specific individuals. ‘reservation_status’ might closely correlate with the target variable, if the target is 'is_canceled', then reservation_status might be a direct indicator, leading to data leakage.

17. Encoding categorical variables

Encoding refers to the process of converting categorical data, which can be in text/string format, into numerical format so that machine learning algorithms can understand and process it.

18. Normalizing numerical variables

a. Reduce skewness: If the data in the columns is right-skewed (i.e., the tail is on the right side), a log transformation can help reduce this skewness and make the data more "normal" or symmetric.

b. Handle wide ranges: Log transformation compresses the scale on which data is represented. If a column has values that span several orders of magnitude, this transformation can make patterns more discernible and reduce the impact of outliers.

c. Stabilize variance: For certain datasets, the variance might increase as the value of a variable increases. Log transformation can stabilize this variance.

d. Easier interpretability: In some scenarios, the log-transformed values (like in the case of money, populations, etc.) might be more interpretable, especially if the original variable grows exponentially.

19. Split the data

In supervised machine learning, it's crucial to assess the performance of a trained model on unseen data. The purpose is to evaluate how well the model generalizes to new, unseen data. Therefore, the data is split into training (70% in this case) and test sets (30%).